Fast multipole method applied to the coupling of elastostatic BEM with FEM

M. Margonari, Marc Bonnet

Fast multipole method applied to elastostatic BEM-FEM coupling

Massimiliano Margonari [1]

Dept. of Mechanical and Structural Engineering, University of Trento, Via Mesiano 77, 38050 Trento, Italy

Marc Bonnet *

Laboratoire de Mécanique des Solides (UMR 7649 CNRS), École Polytechnique, F-91128 Palaiseau cedex, France

Abstract

BEM-FEM coupling is desirable for three-dimensional problems involving specific features such as (i) large or unbounded media with linear constitutive properties, (ii) cracks, (iii) critical parts of complex geometry requiring accurate stress analyses. However, for cases with a BEM discretization involving a large number N_{BEM} of degrees of freedom, setting up the BEM contribution to the coupled problem using conventional techniques is an expensive $O(N_{\mathrm{BEM}}^2)$ task. Moreover, the fully-populated BEM block entails a $O(N_{\mathrm{BEM}}^2)$ storage requirement and a $O(N_{\mathrm{BEM}}^3)$ contribution to the solution time via usual direct solvers. To overcome these pitfalls, the BEM contribution is formulated using the fast multipole method (FMM) and the coupled equations are solved by means of an iterative GMRES solver. Both the storage requirements and the solution times are found to be close to $O(N_{\mathrm{BEM}})$. A preconditioner based on the sparse approximate inverse of the BEM block is shown to improve the convergence of the GMRES solver. Numerical examples involving $N_{\mathrm{BEM}} = O(10^5 - 10^6)$ unknowns, run on a PC computer, are presented; they include the Eshelby inclusion (as a validation example), a many-inclusion configuration, and a dam structure.

Key words: BEM-FEM coupling, Fast multipole method, sparse approximate inverse

1 Introduction

The finite element method (FEM) and the boundary element method (BEM) are important numerical tools for computing the solutions of many engineering problems. FEM is appropriate for very large classes of situations, including e.g.

⋆ *Computers and Strudtures*, **83**:700–717 (2005)

* Corresponding author.

Email addresses: margonar@ing.unitn.it (Massimiliano Margonari), bonnet@lms.polytechnique.fr (Marc Bonnet).

[1] This author was supported in the framework of a research project funded by MIUR (Cofin 2002)

those with heterogeneous or non-linear constitutive properties, or finite deformations. On the other hand, BEM is useful for modelling special situations such as very large or unbounded domains, geometrical singularities (e.g. cracks) or to obtain very accurate results in regions of complicated shape (see e.g. [1, 5, 6]). Coupling the BEM and the FEM allows to exploit their complementary advantages when the geometrical configuration warrants it.

The topic of BEM-FEM coupling has been studied since a long time, and many such coupled formulations have been proposed and analysed [14]. In particular, since the traditional collocation BEM (CBEM) formulations lead to unsymmetric systems of coupled BEM-FEM equations, a number of investigations have been directed towards either forcing the symmetry of the CBEM-FEM equations (like in e.g. [3, 15]), or use a symmetric Galerkin BEM (SGBEM) formulation in order to obtain naturally a symmetric system of BEM-FEM equations (see e.g. [7, 13, 19, 25]). The latter approach is well suited to optimally exploit direct solvers.

As the problem size grows, direct solvers applied to coupled BEM-FEM equations become impractical or infeasible with respect to both computing time and storage, even using specific implementation strategies such as out-of-core procedures, mainly because of the fully-populated nature of the BEM matrices, whose build-up computational cost and storage requirement are both of order $O(N_{\text{BEM}}^2)$, where N_{BEM} denotes the number of degrees of freedom (DOFs) supported by the BEM mesh, not to mention the $O(N_{\text{BEM}}^3)$ growth of the solution time. To overcome these pitfalls, one needs to resort to iterative solution algorithms for linear systems, together with an acceleration technique for computing the BEM contribution to the residual of the matrix BEM-FEM equation.

Coupled SGBEM-FEM formulations usually lead to governing matrices that are symmetric but not sign-definite. In such cases, iterative solvers do not take advantage of the symmetry (in contrast with e.g. the conjugate-gradient technique applied to positive definite problems). Hence the final symmetry of the coupled problem is not as important as in connection with direct solvers. Since CBEM is simpler and less costly to set-up, a good case can be made for considering the unsymmetric CBEM-FEM approach.

In this article, a simple CBEM-FEM coupled approach leading to a system of equations solved by means of the generalized minimal residual (GMRES) iterative algorithm [8, 22] is presented. The BEM part of the calculation is accelerated by means of the Fast Multipole Method (FMM), a method originally introduced by Rokhlin [20] and further discussed in e.g. [10] and in the recent review article by Nishimura [18]. When applied to elastostatic BEM, it provides a reduction of both storage requirements and computational cost to $O(N_{\text{BEM}})$. These improvements make BEM a viable tool (either on a stand-alone basis or coupled with FEM) for large problems. In addition, a preconditioning technique known as the SParse Approximate Inverse (SPAI) technique is implemented for improving the convergence (i.e. reducing the number of iterations) of GMRES. The article is organized as follows. In section 2, CBEM and FEM formulations are outlined and the coupled problem is presented. Then, the FMM treatment of the BEM equations is presented in section 3. The solution technique, and especially the preconditioning strategy, is

discussed in section 4. Finally, numerical examples are examined in section 5.

2 Coupled CBEM-FEM formulation

Consider a solid occupying a three-dimensional region Ω. A coupled BEM-FEM model of the solid (Figure 1) is defined on the basis of a partition $\Omega = \Omega_{\rm B} \cup \Omega_{\rm F}$, where $\partial\Omega_{\rm B}$ (the boundary of $\Omega_{\rm B}$) and $\Omega_{\rm F}$ respectively support boundary element and finite element discretizations. Let $S_{\rm I} = \partial\Omega_{\rm B} \cap \partial\Omega_{\rm F}$ denote the BEM-FEM interface, while $S_{\rm B}$ and $S_{\rm F}$ are the remaining surfaces such that $\partial\Omega_{\rm B} = S_{\rm I} \cup S_{\rm B}$ and $\partial\Omega_{\rm F} = S_{\rm I} \cup S_{\rm F}$. Both subregions $\Omega_{\rm B}, \Omega_{\rm F}$ are here endowed with linear elastic properties:

$$\sigma_{ij} = C_{ijk\ell} u_{k,\ell} \quad (\text{in } \Omega_{\rm B}) \qquad \sigma_{ij} = C^{\rm F}_{ijk\ell} u_{k,\ell} \quad (\text{in } \Omega_{\rm F}) \tag{1}$$

where u_i and σ_{ij} denote the Cartesian components of displacements and stresses and $C_{ijk\ell}, C^{\rm F}_{ijk\ell}$ are the components of the fourth-order elasticity tensors in each region. Besides, homogeneous and isotropic properties are assumed in $\Omega_{\rm B}$, i.e.

$$C_{ijk\ell} = 2\mu \left(\frac{\nu}{1-2\nu} \delta_{ij}\delta_{k\ell} + \delta_{ik}\delta_{j\ell} + \delta_{i\ell}\delta_{jk} \right) \tag{2}$$

(where μ and ν are the shear modulus and the Poisson ratio) whereas the $C^{\rm F}_{ijk\ell}$ need not be subjected to such restrictions. The isotropy assumption in $\Omega_{\rm B}$ is also not mandatory, but is kept here for simplicity and definiteness. The interface need not be simply connected, and $S_{\rm B}$ or $S_{\rm F}$ may be empty (the case of empty $S_{\rm B}$ corresponds to a FEM region embedded in an infinite medium).

CBEM equations. The governing equation for $\Omega_{\rm B}$ is taken here as the usual Somigliana displacement integral equation

$$c_{ij}(\boldsymbol{x}) u_j(\boldsymbol{x}) + \int_{\partial\Omega_{\rm B}} T_{ij}(\boldsymbol{x},\boldsymbol{y}) u_j(\boldsymbol{y}) \, {\rm d}S_y - \int_{\partial\Omega_{\rm B}} U_{ij}(\boldsymbol{x},\boldsymbol{y}) t_j(\boldsymbol{y}) \, {\rm d}S_y = \int_{\partial\Omega_{\rm B}} P_i(\boldsymbol{x},\boldsymbol{y}) \, {\rm d}S_y \tag{3}$$

where $t_i = \sigma_{ij} n_j$ denote the Cartesian components of traction vector defined in terms of the components n_j of the unit normal vector directed away from $\Omega_{\rm B}$.

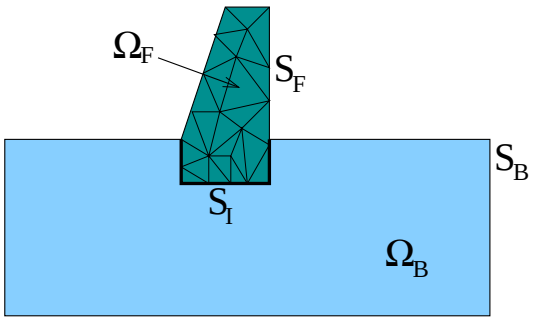

Fig. 1. FEM-BEM coupling: geometry and notation.

The kernels $U_{ij}(\boldsymbol{x},\boldsymbol{y})$ and $T_{ij}(\boldsymbol{x},\boldsymbol{y})$ are the components of the Kelvin fundamental displacement and traction, whose expressions are given later in equations (14) and (17). The last integral with kernel $P_i(\boldsymbol{x},\boldsymbol{y})$ arises when a known gravitational load is considered. The *free-term* $c_{ij}(\boldsymbol{x})$ is $c_{ij}(\boldsymbol{x}) = \delta_{ij}/2$ if $\partial\Omega_{\mathrm{B}}$ is smooth at the *collocation point* $\boldsymbol{x}$, and its value is also known if $\boldsymbol{x}$ is located on an edge or corner of $\partial\Omega_{\mathrm{B}}$ [12, 16]. On introducing isoparametric boundary elements to model geometry and boundary fields, considering in turn all mesh nodes as collocation points, and performing the necessary numerical quadratures [3, 5] and singular integrations using a direct algorithm [12], equation (3) yields the collocation BEM (CBEM) matrix equation

$$[\mathcal{H}]\{u\} - [\mathcal{G}]\{t\} = \{\mathcal{P}\} \tag{4}$$

where $[\mathcal{H}]$ and $[\mathcal{G}]$ are coefficient matrices, $\{u\}$ and $\{t\}$ collect nodal values of boundary displacements and tractions, and $\{\mathcal{P}\}$ contains the contribution of the gravitational load.

FEM equations. In Ω_{F} the standard set of displacement-based FEM discretized equations is considered, i.e.

$$[\mathcal{K}]\{u\} = \{\mathcal{F}\} + \{\mathcal{T}\} \tag{5}$$

where u collects all displacement DOFs in Ω_{F} (including those on S_{I}); $[\mathcal{K}]$ is the elastic stiffness matrix; $\{\mathcal{F}\}$ gathers the nodal generalized forces associated with known loads applied to Ω_{F} (e.g. gravitational body forces, prescribed tractions on S_{F}); and $\{\mathcal{T}\}$ collects the nodal generalized forces associated with tractions along the BEM-FEM interface, i.e. the values of

$$\{\mathcal{T}\} = \left\{ \int_{S_{\mathrm{I}}} t_i(\boldsymbol{y}) \phi^a(\boldsymbol{y}) \, \mathrm{d}S_y \right\} \qquad (1 \leq i \leq 3\,,\ a \in \mathcal{N}_{\mathrm{I}}) \tag{6}$$

where ϕ^a is the trace on S_{I} of the FEM shape function associated with node a of S_{I} and the set $\mathcal{N}_{\mathrm{I}}$ collects the numbers of all FEM nodes lying on S_{I}. The stiffness matrix $[\mathcal{K}]$ is symmetric, positive semidefinite and sparse; it is positive definite if the boundary conditions on S_{F} do not allow any rigid-body motion of Ω_{F} considered in isolation. Here, a skyline storage of $[\mathcal{K}]$ has been used to take advantage of symmetry and sparsity. No attempt at optimizing the storage of $[\mathcal{K}]$ through node renumbering has been made, although this would certainly be useful.

Coupled formulation. Finally, the coupled formulation must include relationships between the displacement and tractions on S_{I} associated with Ω_{B} and Ω_{F}. Here, only the usual perfect bonding condition is considered, i.e.:

$$u_i^{\mathrm{B}}(\boldsymbol{y}) = u_i^{\mathrm{F}}(\boldsymbol{y}) \quad \text{and} \quad t_i^{\mathrm{B}}(\boldsymbol{y}) = t_i^{\mathrm{F}}(\boldsymbol{y}) \qquad (\boldsymbol{y} \in S_{\mathrm{I}}) \tag{7}$$

having conventionally defined the traction vector from both sides of S_{I} in terms of the same unit normal vector, namely that pointing away from Ω_{B}. In the discretized

formulations, these conditions can be imposed in a weak sense through the relations

$$\begin{aligned} \int_{S_{\rm I}} [u_i^{\rm B}(\boldsymbol{y}) - u_i^{\rm F}(\boldsymbol{y})]\phi^a(\boldsymbol{y})\,{\rm d}S_y = 0 \\ \int_{S_{\rm I}} [t_i^{\rm B}(\boldsymbol{y}) - t_i^{\rm F}(\boldsymbol{y})]\phi^a(\boldsymbol{y})\,{\rm d}S_y = 0 \end{aligned} \qquad (a \in \mathcal{N}_{\rm I}) \tag{8}$$

allowing the use of FEM and BEM interpolations whose traces on $S_{\rm I}$ are not conforming, whereas the strong coupling conditions on the nodal values

$$u_i^{\rm B}(\boldsymbol{y}^a) = u_i^{\rm F}(\boldsymbol{y}^a)\ , \quad t_i^{\rm B}(\boldsymbol{y}^a) = t_i^{\rm F}(\boldsymbol{y}^a) \qquad (a \in \mathcal{N}_{\rm I}) \tag{9}$$

require conforming interpolations on $S_{\rm I}$ (interpolations are said to be conforming on $S_{\rm I}$ if the BEM mesh and the trace on $S_{\rm I}$ of the FEM mesh (i) coincide, and (ii) are associated with the same interpolation functions). In this article, the strong coupling conditions (9) are adopted, but weak coupling of type (8) could be easily considered as well.

Gathering equations (4), (5) and (9) leads to the coupled BEM-FEM set of linear equations

$$\begin{bmatrix} \mathcal{A}_{\rm BB} & -\mathcal{G}_{\rm BI} & \mathcal{H}_{\rm BI} & 0 \\ \mathcal{A}_{\rm IB} & -\mathcal{G}_{\rm II} & \mathcal{H}_{\rm II} & 0 \\ 0 & \mathcal{D}_{\rm II} & \mathcal{K}_{\rm II} & \mathcal{K}_{\rm FI} \\ 0 & 0 & \mathcal{K}_{\rm IF} & \mathcal{K}_{\rm FF} \end{bmatrix} \begin{Bmatrix} z_{\rm B} \\ t_{\rm I} \\ u_{\rm I} \\ u_{\rm F} \end{Bmatrix} = \begin{Bmatrix} \mathcal{B}_{\rm B} \\ \mathcal{B}_{\rm I} \\ \mathcal{F}_{\rm I} \\ \mathcal{F}_{\rm F} \end{Bmatrix} \tag{10}$$

having introducing partitions of the DOFs so that $\{t_{\rm I}, u_{\rm I}\}$ gathers the unknown traction and displacement DOFs on $S_{\rm I}$, $\{z_{\rm B}\}$ collects all unknown traction and displacement DOFs on $S_{\rm B}$ (i.e. those *not* prescribed by the boundary conditions), and $\{u_{\rm F}\}$ gathers the displacements at all nodes of $\Omega_{\rm F}$ not lying on $S_{\rm I}$. The number $N_{\rm BEM}$ of unknown DOFs on the BEM mesh (including the BEM-FEM interface) is the cumulated length of $\{z_{\rm B}, t_{\rm I}, u_{\rm I}\}$, while the number $N_{\rm F}$ of *internal* DOFs on the FEM mesh is the length of $\{u_{\rm F}\}$. The first two rows of (10) are obtained by (i) partitioning the CBEM equation (4) according to whether the collocation point $\boldsymbol{x}$ belongs to $S_{\rm B}$ or $S_{\rm I}$, and (ii) performing the usual column-switching according to the boundary conditions on $S_{\rm B}$, with the right-hand sides $\mathcal{B}_{\rm B}, \mathcal{B}_{\rm I}$ gathering all contributions from prescribed data on $S_{\rm B}$. The last two rows of (10) correspond to the subsets of FEM equations obtained by taking the shape function associated with nodes on $S_{\rm I}$ and inside $\Omega_{\rm F}$, respectively, as trial functions. In addition, the strong coupling assumption implies that tractions on $S_{\rm I}$ are modelled using the shape functions ϕ^a associated with the BEM discretization of $S_{\rm I}$, so that the load vector $\{\mathcal{T}\}$ in (5) becomes (in terms of the above partition) $\{\mathcal{T}\} = [\mathcal{D}_{\rm II}]\{t_{\rm I}\}$, where the entries of the matrix $[\mathcal{D}_{\rm II}]$ are

$$[\mathcal{D}_{\rm II}] = \left[\delta_{ij} \int_{S_{\rm I}} \phi^a(\boldsymbol{y})\phi^b(\boldsymbol{y})\,{\rm d}S_y\right] \qquad (1 \leq i,j \leq 3\ ,\ (a,b) \in \mathcal{N}_{\rm I} \times \mathcal{N}_{\rm I}) \tag{11}$$

Solution strategy. In the system (10), the blocks generated by the CBEM equations are fully populated and nonsymmetric, whereas the blocks contributed by the

FEM equations are sparse and symmetric. For problems of moderate size, direct solvers may be applied to the system (10). This usually entails some sort of condensation, in order not to build explicitly the whole matrix (which features blocks of zeros). Alternative BEM-FEM formulations based on the symmetric Galerkin BEM instead of the CBEM lead to symmetric variants of the system (10).

As the problem size grows, direct solvers become impractical or infeasible with respect to both computing time and storage, mainly due to the fully-populated nature of the BEM blocks $[\mathcal{G}]$ and $[\mathcal{H}]$, and iterative solvers are used instead. In the present case of non-symmetric coupled equations, the system (10) is solved using the generalized minimal residual (GMRES) algorithm [8, 22], which is applicable to general invertible square matrices. Such algorithms are based on matrix-vector evaluations, and therefore do not require actual storage of the matrix in (10).

In this context, traditional quadrature methods mean that each evaluation of the BEM contribution to the matrix-vector product in (10), i.e. of the residuals

$$\begin{aligned} &[\mathcal{A}_{\text{BB}}]\,\{z_{\text{B}}\} - [\mathcal{G}_{\text{BI}}]\{t_{\text{I}}\} + [\mathcal{H}_{\text{BI}}]\{u_{\text{I}}\} \\ &[\mathcal{A}_{\text{IB}}]\,\{z_{\text{B}}\} - [\mathcal{G}_{\text{II}}]\{t_{\text{I}}\} + [\mathcal{H}_{\text{II}}]\{u_{\text{I}}\} \end{aligned} \tag{12}$$

is a $O(N_{\text{BEM}}^2)$ task, and BEM numerical quadratures are usually time-consuming (i.e. the coefficient of $O(N_{\text{BEM}}^2)$ is expected to be fairly large). To accelerate the computation of the residuals (12), the Fast Multipole Method (FMM) is adopted. In the present context of elastostatics, choosing this methodology allows to reduce the computational burden of this task to $O(N_{\text{BEM}})$. Detailed descriptions of FMM-based algorithms can be found in [9, 10, 18]. Besides, the FEM stiffness matrix $[\mathcal{K}]$ is evaluated only once, stored (in sky-line fashion) and then invoked at every GMRES iteration. The computation of the sparse matrix $[\mathcal{D}_{\text{II}}]$ is very inexpensive and is therefore done for every GMRES iteration (i.e. $[\mathcal{D}_{\text{II}}]$ is not stored).

The methodology and formulae pertaining to FMM applied to the Somigliana integral equation (3) are concisely reviewed in section 3. Then, section 4 will address the preconditioning strategy used in connection with the GMRES algorithm.

3 Fast Multipole Method (FMM)

3.1 Multipole expansions for 3D elasticity

The FMM used in this article follows closely the treatment presented in [24] for solving the hypersingular CBEM for elastostatic crack problems. The latter is based on the following series expressing the inverse of the distance $r = |\boldsymbol{y} - \boldsymbol{x}|$ between two points $\boldsymbol{y}$ and $\boldsymbol{x}$, expanded about two poles $\boldsymbol{y}^0$ and $\boldsymbol{x}^0$:

$$\frac{1}{r} = \sum_{n=0}^{+\infty} \sum_{m=-n}^{n} R_{n,m}(\hat{\boldsymbol{y}}) \sum_{n'=0}^{+\infty} \sum_{m'=-n'}^{n'} (-1)^n \overline{S_{n+n',m+m'}(\boldsymbol{r}^0)} R_{n',m'}(\hat{\boldsymbol{x}}) \tag{13}$$

having put $\hat{\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{y}^0$, $\hat{\boldsymbol{x}} = \boldsymbol{x} - \boldsymbol{x}^0$ and $\boldsymbol{r}^0 = \boldsymbol{y}^0 - \boldsymbol{x}^0$, and where the overbar indicates complex conjugation. The (complex-valued) *solid harmonics* $R_{n,m}(\boldsymbol{z})$,

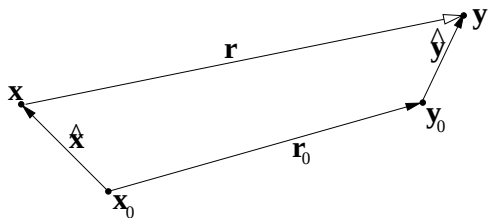

Fig. 2. Geometrical notation for the multipole expansion.

$S_{n,m}(\boldsymbol{z})$ can be evaluated by means of the recursive formulae (A.1) to (A.5) given in the appendix. For fixed poles $\boldsymbol{y}^0$ and $\boldsymbol{x}^0$, expansion (13) is convergent for all $\boldsymbol{y}$ and $\boldsymbol{x}$ such that $|\hat{\boldsymbol{y}}| < |\boldsymbol{x} - \boldsymbol{y}^0|$ and $|\hat{\boldsymbol{x}}| < |\boldsymbol{y} - \boldsymbol{x}^0|$ (Figure 2).

To take advantage of expansion (13), it is convenient to formulate the elastostatic Kelvin fundamental displacement in the form

$$U_{ij}(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{2\mu A}\Big[(3-4\nu)\delta_{ij}\frac{1}{r} - (\hat{x}_j - r_j^0 - \hat{y}_j)\frac{\partial}{\partial x_i}\frac{1}{r}\Big] \tag{14}$$

where $A = 8\pi(1-\nu)$. Introducing (13) into the above formula yields the series expansion

$$U_{ij}(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{2\mu A}\sum_{n=0}^{+\infty}\sum_{m=-n}^{n} F_{ij}^{m,n}(\hat{\boldsymbol{x}}-\boldsymbol{r}^0)R_{n,m}(\hat{\boldsymbol{y}}) + G_i^{m,n}(\hat{\boldsymbol{x}}-\boldsymbol{r}^0)\hat{y}_j R_{n,m}(\hat{\boldsymbol{y}}) \tag{15}$$

having put

$$\begin{aligned} F_{ij}^{m,n}(\hat{\boldsymbol{x}}-\boldsymbol{r}^0) &= \sum_{n'=0}^{+\infty}\sum_{m'=-n'}^{n'} (-1)^n \overline{S_{n+n',m+m'}(\boldsymbol{r}^0)} \\ &\qquad \times \Big[(3-4\nu)\delta_{ij}R_{n',m'}(\hat{\boldsymbol{x}}) - (\hat{x}_j - r_j^0)\frac{\partial}{\partial x_i}R_{n',m'}(\hat{\boldsymbol{x}})\Big] \qquad (16) \\ G_i^{m,n}(\hat{\boldsymbol{x}}-\boldsymbol{r}^0) &= \sum_{n'=0}^{+\infty}\sum_{m'=-n'}^{n'} (-1)^n \overline{S_{n+n',m+m'}(\boldsymbol{r}^0)}\frac{\partial}{\partial x_i}R_{n',m'}(\hat{\boldsymbol{x}}) \end{aligned}$$

Then, the Kelvin traction vector $T_{ij}(\boldsymbol{x},\boldsymbol{y})$ is defined by

$$T_{ij}(\boldsymbol{x},\boldsymbol{y}) = C_{jpk\ell}n_p(\boldsymbol{y})\frac{\partial}{\partial y_\ell}U_{ik}(\boldsymbol{x},\boldsymbol{y}) \tag{17}$$

and therefore admits the series representation

$$\begin{aligned} T_{ij}(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{2\mu A}C_{jpk\ell}n_p(\boldsymbol{y})\sum_{n=0}^{+\infty}\sum_{m=-n}^{n}\Big\{ & F_{ik}^{m,n}(\hat{\boldsymbol{x}}-\boldsymbol{r}^0)\frac{\partial}{\partial y_\ell}R_{n,m}(\hat{\boldsymbol{y}}) \\ & + G_i^{m,n}(\hat{\boldsymbol{x}}-\boldsymbol{r}^0)\frac{\partial}{\partial y_\ell}\Big(\hat{y}_k R_{n,m}(\hat{\boldsymbol{y}})\Big)\Big\} \end{aligned} \tag{18}$$

The iterative GMRES solver requires repeated evaluations of the residual of the displacement integral equation (3) for known trial solutions (u_j, t_j). With that

purpose in mind, let the surface $\partial\Omega_{\mathrm{B}}$ be split into two complementary parts $S(\boldsymbol{x}^0)$ and $\bar{S}(\boldsymbol{x}^0)$, where $S(\boldsymbol{x}^0)$ contains $\boldsymbol{x}^0$ as an interior point. The residual of equation (3) for a trial solution (u_j, t_j) is recast in the form

$$c_{ij}(\boldsymbol{x})u_j(\boldsymbol{x}) + \int_{S(x^0)} T_{ij}(\boldsymbol{x},\boldsymbol{y})u_j(\boldsymbol{y})\,\mathrm{d}S_y - \int_{S(x^0)} U_{ij}(\boldsymbol{x},\boldsymbol{y})t_j(\boldsymbol{y})\,\mathrm{d}S_y + \bar{\mathcal{I}}_i(\boldsymbol{x}) \quad (19)$$

having omitted for simplicity the body force term and where $\bar{\mathcal{I}}_i(\boldsymbol{x})$, defined by

$$\bar{\mathcal{I}}_i(\boldsymbol{x}) = \int_{\bar{S}(x^0)} U_{ij}(\boldsymbol{x},\boldsymbol{y})t_j(\boldsymbol{y})\,\mathrm{d}S_y - \int_{\bar{S}(x^0)} T_{ij}(\boldsymbol{x},\boldsymbol{y})u_j(\boldsymbol{y})\,\mathrm{d}S_y \quad (20)$$

collects the far-field contributions relative to $\boldsymbol{x}$ or $\boldsymbol{x}^0$. With respect to decomposition (19), *near-field* element integrals (i.e. those for elements lying in $S(\boldsymbol{x}^0)$) are to be evaluated by means of the usual numerical integration techniques classically used for BEMs [3, 5], while the *far-field* contribution $\bar{\mathcal{I}}_i(\boldsymbol{x})$ will be evaluated with the help of the multipole expansions. Accordingly, decomposition (19) is only considered for collocation points $\boldsymbol{x}$ lying inside $S(\boldsymbol{x}^0)$ in such a way that $S(\boldsymbol{x}^0)$ includes the support of all singular element integrals. The direct algorithm [12] is chosen here for the evaluation of the latter, because decomposition (19) is not well suited to using an indirect regularization based on a rigid-body identity.

On introducing the series representations (15) and (18) of the fundamental kernels into (20), $\bar{\mathcal{I}}_i(\boldsymbol{x})$ takes the form

$$\bar{\mathcal{I}}_i(\boldsymbol{x}) = \frac{1}{2\mu A}\sum_{n=0}^{+\infty}\sum_{m=-n}^{n}\Big\{F_{ij}^{m,n}(\hat{\boldsymbol{x}}-\boldsymbol{r}^0)\Big[M^{\mathrm{t1}}_{j;n,m}(\boldsymbol{y}^0) - M^{\mathrm{u1}}_{j;n,m}(\boldsymbol{y}^0)\Big] + G_i^{m,n}(\hat{\boldsymbol{x}}-\boldsymbol{r}^0)\Big[M^{\mathrm{t2}}_{n,m}(\boldsymbol{y}^0) - M^{\mathrm{u2}}_{n,m}(\boldsymbol{y}^0)\Big]\Big\} \quad (21)$$

where the *multipole moments* are defined by

$$\begin{aligned}
M^{\mathrm{t1}}_{j;n,m}(\boldsymbol{y}^0) &= \int_{\bar{S}(x^0)} R_{n,m}(\hat{\boldsymbol{y}})t_j(\boldsymbol{y})\,\mathrm{d}S_y \\
M^{\mathrm{t2}}_{n,m}(\boldsymbol{y}^0) &= \int_{\bar{S}(x^0)} \hat{y}_j R_{n,m}(\hat{\boldsymbol{y}})t_j(\boldsymbol{y})\,\mathrm{d}S_y \\
M^{\mathrm{u1}}_{j;n,m}(\boldsymbol{y}^0) &= C_{kpj\ell}\int_{\bar{S}(x^0)} \frac{\partial}{\partial y_\ell}R_{n,m}(\hat{\boldsymbol{y}})u_k(\boldsymbol{y})n_p(\boldsymbol{y})\,\mathrm{d}S_y \\
M^{\mathrm{u2}}_{n,m}(\boldsymbol{y}^0) &= C_{kpj\ell}\int_{\bar{S}(x^0)} \frac{\partial}{\partial y_\ell}\Big(\hat{y}_j R_{n,m}(\hat{\boldsymbol{y}})\Big)u_k(\boldsymbol{y})n_p(\boldsymbol{y})\,\mathrm{d}S_y
\end{aligned} \quad (22)$$

Further, introducing expressions (16) into (21), rearranging summations by switching dummy summation indices (m, n) and (m', n'), $\bar{\mathcal{I}}_i(\boldsymbol{x})$ takes the form

$$\bar{\mathcal{I}}_i(\boldsymbol{x}) = \frac{1}{2\mu A}\sum_{n=0}^{+\infty}\sum_{m=-n}^{n}\Big\{\Big[(3-4\nu)\delta_{ij}R_{n,m}(\hat{\boldsymbol{x}}) - (\hat{x}_j - r_j^0)\frac{\partial}{\partial x_i}R_{n,m}(\hat{\boldsymbol{x}})\Big]L^1_{j;n,m}(\boldsymbol{x}^0) + \frac{\partial}{\partial x_i}R_{n,m}(\hat{\boldsymbol{x}})L^2_{j;n,m}(\boldsymbol{x}^0)\Big\} \quad (23)$$

where the *local expansions* are related to the multipole moments by the following "multipole-to-local" (M2L) relations:

$$\begin{aligned} L^1_{j;n,m}(\boldsymbol{x}^0) &= \sum_{n'=0}^{+\infty}\sum_{m'=-n'}^{n'} (-1)^{n'}\overline{S_{n+n',m+m'}(\boldsymbol{r}^0)}\Big[M^{\mathrm{t1}}_{j;n',m'} - M^{\mathrm{u1}}_{j;n',m'}\Big](\boldsymbol{y}^0) \\ L^2_{n,m}(\boldsymbol{x}^0) &= \sum_{n'=0}^{+\infty}\sum_{m'=-n'}^{n'} \overline{S_{n+n',m+m'}(\boldsymbol{r}^0)}\Big[M^{\mathrm{t2}}_{n',m'} - M^{\mathrm{u2}}_{n',m'}(\boldsymbol{y}^0)\Big](\boldsymbol{y}^0) \end{aligned} \tag{24}$$

3.2 Outline of FMM algorithm

Traditional methods for evaluating the residual of integral equation (3) require that all element integrals be recomputed each time a new collocation point $\boldsymbol{x}$ is considered, hence the $O(N^2_{\mathrm{BEM}})$ overall complexity of such operation. In contrast, equation (23) shows that the *same* set of local expansions $L^1_{j;n,m}(\boldsymbol{x}^0)$, $L^2_{j;n,m}(\boldsymbol{x}^0)$ allows the treatment of not one but a cluster of collocation points lying sufficiently close to a pole $\boldsymbol{x}^0$.

If there is a positive number R such that $\hat{\boldsymbol{x}} < R$ and $|\boldsymbol{r}^0 + \hat{\boldsymbol{y}}| > 2R$, the truncation error estimate

$$\left|\frac{1}{r} - \sum_{n=0}^{p}\sum_{m=-n}^{n} R_{n,m}(\hat{\boldsymbol{y}}) \sum_{n'=0}^{n}\sum_{m'=-n'}^{n'} \overline{S_{n+n',m+m'}(\boldsymbol{r}^0)} R_{n',m'}(\hat{\boldsymbol{x}})\right| < \frac{1}{2^{p+1}R} \tag{25}$$

for the multipole expansion (13) [10] implies that the larger the radius of the cluster of collocation points, the smaller the size of the far-field surface $\bar{S}(\boldsymbol{x}^0)$.

To exploit optimally the acceleration afforded by (23), a hierarchical oct-tree structure of elements is introduced. For that purpose, a cube containing the whole boundary $\partial\Omega_{\mathrm{B}}$, called 'level-0 cell', is divided into eight cubes (level-1 cells), each of which is divided in the same fashion. A level-ℓ cell is divided into level-$(\ell+1)$ cells unless it contains less than a preset (relatively small) number M of boundary elements (such cells are termed *leaves*). The FMM algorithm then consists of:

- An *upward pass* where multipole moments (22) are first computed for the lowest-level cells and then recursively aggregated by moving upward in the tree until level 2 (for which there are $4 \times 4 \times 4$ cells overall) is reached. This operation requires the so-called "multipole-to-multipole" (M2M) identities

$$\begin{aligned} M^{(\mathrm{u,t})1}_{j;n,m}(\boldsymbol{y}^0) &= \sum_{n'=0}^{n}\sum_{m'=-n'}^{n'} R_{n',m'}(\boldsymbol{y}^{0\prime} - \boldsymbol{y}^0) M^{(\mathrm{u,t})1}_{j;n-n',m-m'}(\boldsymbol{y}^{0\prime}) \\ M^{(\mathrm{u,t})2}_{n,m}(\boldsymbol{y}^0) &= \sum_{n'=0}^{n}\sum_{m'=-n'}^{n'} R_{n',m'}(\boldsymbol{y}^{0\prime} - \boldsymbol{y}^0)\Big[M^{(\mathrm{u,t})2}_{n-n',m-m'}(\boldsymbol{y}^{0\prime}) \\ &\qquad - (y^{0\prime}_j - y^0_j) M^{(\mathrm{u,t})1}_{j;n-n',m-m'}(\boldsymbol{y}^{0\prime})\Big] \end{aligned} \tag{26}$$

  to shift the origin from the center $\boldsymbol{y}^{0\prime}$ of a level-$(\ell+1)$ cell to the center $\boldsymbol{y}^0$ of a level-ℓ cell.

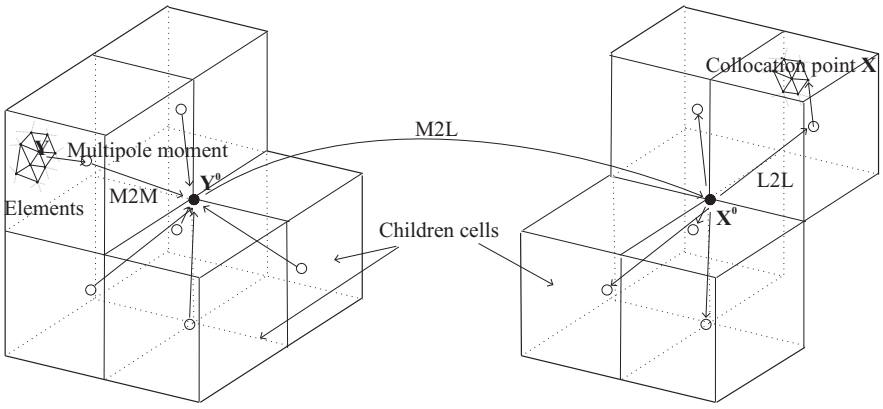

Fig. 3. M2M, M2L and L2L translations

- A *downward pass* where local expansions are first computed at level $\ell = 2$ and then evaluated at selected lower-level cells by tracing the tree structure downwards. This operation requires the so-called "local-to-local" (L2L) identities

$$
\begin{aligned}
L^1_{j;n,m}(\boldsymbol{x}^{0\prime}) &= \sum_{n'=0}^{n} \sum_{m'=-n'}^{n'} R_{n-n',m-m'}(\boldsymbol{x}^{0\prime} - \boldsymbol{x}^0) L^1_{j;n',m'}(\boldsymbol{x}^0) \\
L^2_{n,m}(\boldsymbol{x}^{0\prime}) &= \sum_{n'=0}^{n} \sum_{m'=-n'}^{n'} R_{n-n',m-m'}(\boldsymbol{x}^{0\prime} - \boldsymbol{x}^0) \\
&\qquad \left[L^2_{n',m'}(\boldsymbol{x}^0) - (x^{0\prime}_j - x^0_j) L^1_{j;n',m'}(\boldsymbol{x}^0) \right]
\end{aligned}
\tag{27}
$$

  to shift the origin from the center $\boldsymbol{x}^0$ of a level-(ℓ) cell to the center $\boldsymbol{x}^{0\prime}$ of a level-$(\ell+1)$ cell. At all levels, only interaction between well-separated cells are so evaluated.
- A direct calculation, which includes all singular integrations, where the near-field contributions to (19) are evaluated using conventional integration methods.

For BEM formulations of static problems, one computation of the residual of integral equation (3) the FMM is found to be a $O(N_{\text{BEM}})$ task.

The near-field integrations, as they involve all singular and nearly-singular integrations, are a costly (albeit $O(N_{\text{B}})$) task. For that reason, they were not recomputed for each iteration (i.e. each evaluation of the residual (12)) in the present implementation. Instead, that part of the matrix block is precomputed and assembled into a matrix, prior to invoking GMRES. The near-field direct influence matrix thus obtained is sparse, with $O(N_{\text{BEM}})$ entries, and is of course stored accordingly.

4 Preconditioning strategy

As previously mentioned, the GMRES iterative algorithm is used for solving the system (10) of CBEM-FEM equations. The convergence rate of iterative solvers depends strongly on the spectral properties of the coefficient matrix. These spectral properties can be improved by means of suitably chosen linear transformations,

i.e. *preconditioning*. Preconditioning iterative solvers is thus an important practical issue, to which a great deal of attention is devoted in the literature [8, 21].

The linear system in equation (10) has the form

$$[A]\{X\} = \{B\} \tag{28}$$

where the $N \times N$ matrix $[A]$ is invertible but not symmetric. Preconditioning of (28) can take any of the forms

$$\begin{array}{lll} [P^{\mathrm{L}}][A]\{X\} = [P^{\mathrm{L}}]\{B\} & & \text{(left)} \\ [A][P^{\mathrm{R}}]\{Y\} = \{B\}\,, & \{X\} = [P^{\mathrm{R}}]\{Y\} & \text{(right)} \\ [P^{\mathrm{L}}][A][P^{\mathrm{R}}]\{Y\} = [P^{\mathrm{L}}]\{B\}\,, & \{X\} = [P^{\mathrm{R}}]\{Y\} & \text{(two-sided)} \end{array} \tag{29}$$

where the *left preconditioner* $[P^{\mathrm{L}}]$ or the *right preconditioner* $[P^{\mathrm{R}}]$ might ideally be chosen so that the governing matrix of the preconditioned system of equations is the identity matrix. Of course, this would amount to solve the original system by means of a direct solver, and therefore be of little practical value. A good preconditioning strategy is such that the matrix of the resulting linear system of equations is as close as possible to the identity matrix. This objective must in addition be fulfilled within computing time and storage requirements significantly lower, and growing slower with N, than those expected for the original system of equations.

Preconditioners are either implicit or explicit. The former require the solution of a linear system for each step of the iterative procedure. The most popular are based on an incomplete LU decomposition (ILU or ILUD) of the original matrix or a sparsified version thereof, whereas the latter are based on an explicit evaluation of the preconditioning matrices $[P^{\mathrm{L}}]$ and/or $[P^{\mathrm{R}}]$. Such matrices are then usually chosen to be sparse so as to reduce the storage requirement and the computational complexity of the preconditioner evaluation.

In this article, a two-sided preconditioning of equation (10) has been adopted, with the left and right preconditioning matrices defined by

$$P^{\mathrm{L}} = \begin{bmatrix} & \mathcal{S} & & 0 \\ & & & 0 \\ 0 & 0 & \mathrm{Id} & 0 \\ 0 & 0 & 0 & \mathcal{C}^{-\mathrm{T}} \end{bmatrix} \qquad P^{\mathrm{R}} = \begin{bmatrix} \mathrm{D}(\mathcal{A}_{\mathrm{BB}}) & 0 & 0 & 0 \\ 0 & \mathrm{D}(-\mathcal{G}_{\mathrm{II}}) & 0 & 0 \\ 0 & 0 & \mathrm{D}(\mathcal{K}_{\mathrm{II}}) & 0 \\ 0 & 0 & 0 & \mathcal{C}^{-1} \end{bmatrix} \tag{30}$$

where $\mathrm{D}(X)$ denotes the inverse of the main diagonal of X, Id is the identity block and $\mathcal{K}_{\mathrm{FF}} = \mathcal{C}\mathcal{C}^{\mathrm{T}}$ denotes the Choleski decomposition of $\mathcal{K}_{\mathrm{FF}}$. The block $\mathcal{S}$ is constructed by computing, in a manner explained later in this section, the sparse approximate inverse (SPAI) of the $N_{\mathrm{BEM}} \times N_{\mathrm{BEM}}$ matrix

$$\begin{bmatrix} \mathcal{A}_{\mathrm{BB}} & -\mathcal{G}_{\mathrm{BI}} & \mathcal{H}_{\mathrm{BI}} \\ \mathcal{A}_{\mathrm{IB}} & -\mathcal{G}_{\mathrm{II}} & \mathcal{H}_{\mathrm{II}} \\ 0 & 0 & \mathrm{Diag}(\mathcal{K}_{\mathrm{II}}) \end{bmatrix} \tag{31}$$

and keeping only the first two rows (in the above block notation sense) of the result. The blocks $\mathcal{C}$ and $\mathcal{C}^{\mathrm{T}}$ clearly define an implicit preconditioning on $\{u_{\mathrm{F}}\}$, while the preconditioning acting on $\{z_{\mathrm{B}}, t_{\mathrm{I}}, u_{\mathrm{I}}\}$ is of explicit type.

As mentioned above, the block $\mathcal{S}$ in (30) results from the computation of a sparse approximate inverse (SPAI). The concept of SPAI is described in e.g. [4, 11, 21], where in particular theoretical properties and invertibility of such preconditioning matrices are discussed. This technique appears to be effective even when applied to unsymmetric matrices and with an imposed sparsity of 97-99%. An application of the SPAI preconditioning technique to a BEM formulation for electromagnetic wave problems is presented in [2].

The left sparse approximate inverse $[A^\sharp]$ of a generic square matrix $[A]$ is defined as the solution of the minimization problem

$$[A^\sharp] = \underset{[E]}{\arg\min} \parallel [I] - [E][A] \parallel_{\mathrm{F}}^2 \qquad [E] \in \mathbb{R}^{N\times N} \text{ sparse} \tag{32}$$

where $\|[E]\|_{\mathrm{F}} = (E_{ab}E_{ab})^{1/2}$ denotes the Frobenius norm and the sparsity constraint may either take the form of a predetermined sparsity pattern or be found *a posteriori* so as to satisfy some supplementary optimality criterion. The former, obviously simpler, approach is adopted here. The choice of the Frobenius norm allows to decompose (32) into N independent minimization problems for the rows of $[A^\sharp]$:

$$\{A_k^\sharp\} = \underset{\{E\}}{\arg\min} \parallel \{e_k\} - \{E\}[A] \parallel \qquad \{E\} \in \mathbb{R}^{1\times N} \text{ sparse} \tag{33}$$

where $\{e_k\}$ and $\{A_k^\sharp\}$ are the k^{th} row of the identity matrix and $[A^\sharp]$ respectively, and the trial row N-vector $\{E\}$ is constrained by the preset sparsity pattern assigned to the k-th row of $[A^\sharp]$.

The effectiveness of SPAI-based preconditioners largely depends on the choice of sparsity pattern for $[A^\sharp]$. This choice should be such that $[A^\sharp]$ provides a good approximation of the largest entries in $[A^{-1}]$. A difficulty arises in that the location of such entries in $[A^{-1}]$ is usually not known *a priori*. In [11] the sparsity pattern is found for every row (or column) with an adaptive strategy, and the positions of non-zero entries in $[A^\sharp]$ are iteratively modified to minimize the residual in (33). In [2], four heuristics for selecting *a priori* the sparsity pattern of $[A^\sharp]$ are considered.

Here, the sparsity pattern is predetermined by means of the following strategy. The number $m \ll N$ of nonzero entries in each row of $[A^\sharp]$ is chosen. A sparse approximation $[\hat{A}]$ of $[A]$ is created by assigning to $[\hat{A}]$ the m largest entries of each row of $[A]$, at the same locations, and setting all other entries of $[\hat{A}]$ to zero. The sparsity pattern of $[\hat{A}]$ thus defined is then assigned to $[A^\sharp]$ as well, and the entries of $[A^\sharp]$ are found by solving the minimization problems (33) with $[A]$ replaced with $[\hat{A}]$. Denoting by $\mathcal{I}(k)$ the set of the m column indices of nonzero entries in the k-th row of $[A^\sharp]$ or $[\hat{A}]$, the independent minimization prolems are found to reduce to the minimization of quadratic functionals over m-vectors:

$$\{\hat{A}_i^\sharp\} = \underset{\{\hat{E}\}\in\mathbb{R}^{1,m}}{\arg\min} \left\{ \parallel \{E\}[\tilde{A}_i] \parallel^2 - 2\,\mathrm{trace}(\{E\}[\tilde{A}_i]) + 1 \right\} \qquad (1 \le i \le N) \tag{34}$$

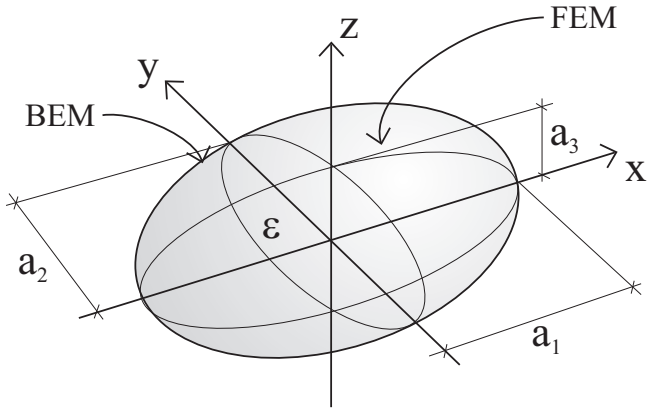

Fig. 4. Inclusion problem: geometry and notation.

where $\{\hat{A}_i^\sharp\}$ is the m-vector of the nonzero entries of row i of $[A^\sharp]$ and $[\tilde{A}_i]$ is the $m \times m$ matrix with entries $[A_{ab}]_{a\in\mathcal{I}(i),b\in\mathcal{I}(a)}$. The N minimization problems (34) are of small size, allowing for a relatively inexpensive set-up of $[A^\sharp]$. Computing $[A^\sharp]$ is therefore a $O(m^3N)$ task, while storing $[A^\sharp]$ requires a $O(mN) \ll N^2$ memory space. They are well-posed provided the diagonal of $[A]$ is included in the sparsity pattern of $[A^\sharp]$, i.e. that $k \in \mathcal{I}(k)$ for every k.

As previously mentioned, the technique outlined above is used for computing the sparse approximate inverse of the matrix defined by (31), from which the block $\mathcal{S}$ of (30) is then extracted.

5 Numerical examples

Two numerical examples based on the CBEM-FEM coupling method described above are presented in this section. The first one (section 5.1), which concerns the response of an infinite medium to a uniform temperature applied over an ellipsoidal region, has an exact solution. The second one (section 5.2) concerns a real dam structure, in order to demonstrate the present coupling technique on a realistic problem. For both examples, the stopping criterion for the GMRES algorithm was a backward error less than 10^{-5}, i.e.: $\| \{B\} - [A]\{X\} \| / \| \{B\} \| \leq 10^{-5}$ with the notations of equation (10). All computations were run on a Linux PC computer equipped with one 3 GHz Pentium 4 CPU unit and 2 GBytes of core memory.

5.1 Eshelby solution

This first numerical example involves an infinite elastic body containing an ellipsoidal inclusion $\mathcal{E}$, with principal half axes a_1, a_2, a_3 (Figure 4). The Cartesian coordinate system is chosen such that the origin coincides with the center of $\mathcal{E}$ while the coordinate axes are directed along the principal axes of $\mathcal{E}$. The inclusion and the surrounding medium have the same elastic moduli (μ, ν); in addition, $\mathcal{E}$ is endowed with a constant thermal expansion coefficient α. The elastic response of the infinite body to a *uniform* temperature increment $\Delta T^\star$ applied over $\mathcal{E}$ is such that the stress

Table 1
Inclusion problem: meshes, oct-tree parameters and numbers of DOFs.

Mesh	Nodes	Elements		Oct-tree		DOFs		
		BEM	FEM	Max level	Leaves	N_{BEM}	N_{F}	$N = N_{\text{BEM}}+N_{\text{F}}$
1	267	346	979	3	42	1050	276	1326
2	822	1038	3153	3	100	3126	903	4029
3	1362	1540	5563	3	103	4632	1770	6402
4	2274	2418	9626	4	301	7266	3189	10455
5	5881	5200	26602	4	422	15612	9837	25449
6	12868	9402	61770	5	1175	28218	24495	52713
7	20258	12842	100200	6	1403	38538	41505	80043

inside the inclusion is constant. The exact solution for the stress inside $\mathcal{E}$, a special case of the well-known Eshelby solution (see e.g. [17]), is known. The nonzero components of the stress inside $\mathcal{E}$ are given by

$$\sigma_{ii} = \frac{E}{1-\nu}(I_i - 1)\alpha\Delta T^\star \qquad (i = 1, 2, 3) \tag{35}$$

(no summation over i) with

$$I_i = \frac{a_1 a_2 a_3}{2}\int_0^\infty \frac{\mathrm{d}s}{(a_i^2 + s)\Delta(s)}$$

and $\Delta(s) = (a_1^2 + s)^{1/2}(a_2^2 + s)^{1/2}(a_3^2 + s)^{1/2}$.

This example is used here to demonstrate some of the computational features of the present BEM-FEM coupling method. The FEM domain is taken as the inclusion, i.e. $\Omega_{\text{F}} = \mathcal{E}$, and the BEM-FEM interface is therefore $\partial\mathcal{E}$, the boundary of the inclusion. Although in principle this example could be treated by taking advantage of the symmetries with respect to the three coordinate planes, the entire domain has

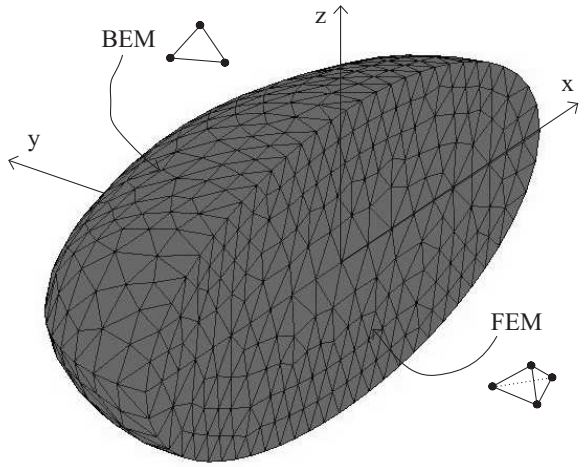

Fig. 5. Inclusion problem: longitudinal section ((x, z)-plane) of mesh 1.

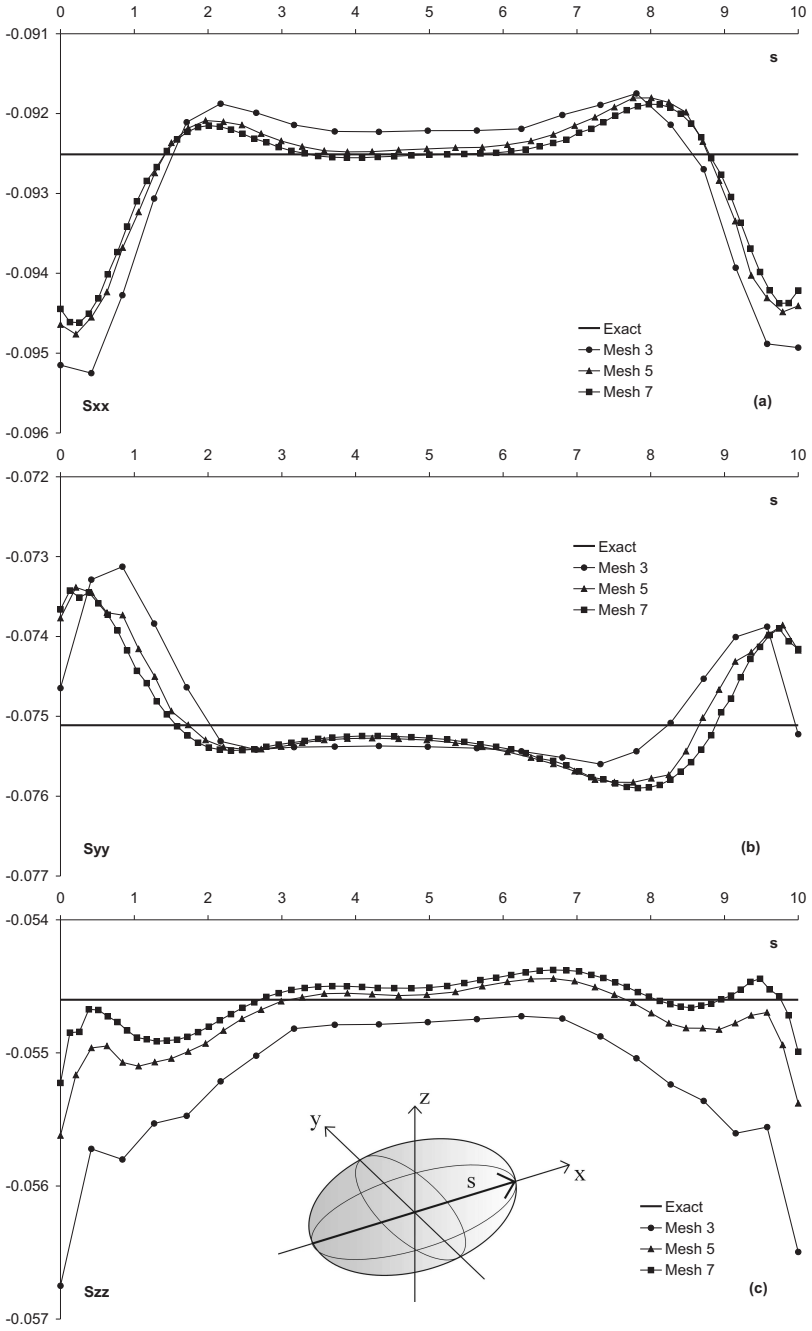

Fig. 6. Inclusion problem: non-zero stress components along the major x-axis within the inclusion.

been modeled. The physical parameters used in the numerical examples, in consistent SI units, are $E = 2\mu(1+\nu) = 100$, $\nu = 0.1$, $\alpha = 10^{-4}$, $(a_1, a_2, a_3) = (5, 3, 2)$, $\Delta T^{\star} = 10$. The interface $S_{\rm I}$ and the FEM region $\Omega_{\rm F}$ have been modelled using seven meshes (Figure 5), involving DOFs for displacements $u_{\rm I}$ and tractions $t_{\rm I}$ at the interface nodes, and for displacements $u_{\rm F}$ at the internal FEM nodes (i.e. excluding the interface nodes). The BEM and FEM interpolations are based on linear three-noded boundary triangular elements and linear four-noded tetrahedral domain elements, respectively. The problem sizes N range between about 1300 (mesh 1)

Table 2
Inclusion problem: CPU timing using SPAI-based preconditioning.

Mesh	Precond. (s)		Time (s)				Iters	Total time
	BEM	FEM	Upw.	Downw.	Direct	Cycle	n	(s)
1	10	<1	1	<1	<1	1	37	43
2	36	<1	1	1	<1	2	37	154
3	50	<1	1	2	<1	3	37	202
4	64	3	3	2	1	6	36	277
5	169	18	7	3	1	11	37	721
6	349	101	12	6	2	19	38	1425
7	512	279	18	6	18	42	38	2913

to about 80 000 (mesh 7). Details on the meshes, oct-tree parameters and numbers of DOFs are provided in Table 1.

In Figure 6 the non-zero stress components (σ_{11}, σ_{22} and σ_{33} evaluated inside $\mathcal{E}$ and along the x-axis are plotted against the exact solution (35). The agreement is reasonable, with relative errors always less than 3%. The comparatively high errors recorded near the interface even with fine meshes have to be ascribed to the FEM and not to the coupling technique or to the FMM. The same results have been in fact obtained with the traditional CBEM approach.

Table 2 displays for each of the seven meshes the computer times spent in (i) the computation of the preconditioning blocks associated with the boundary element and finite element parts of the system (10), (ii) the BEM matrix-vector products performed using FMM for each iteration, and (iii) the overall analysis. In particular, the times recorded for setting up the BEM-based part of the preconditioner include the computation of the SPAI as explained in section 4. The term

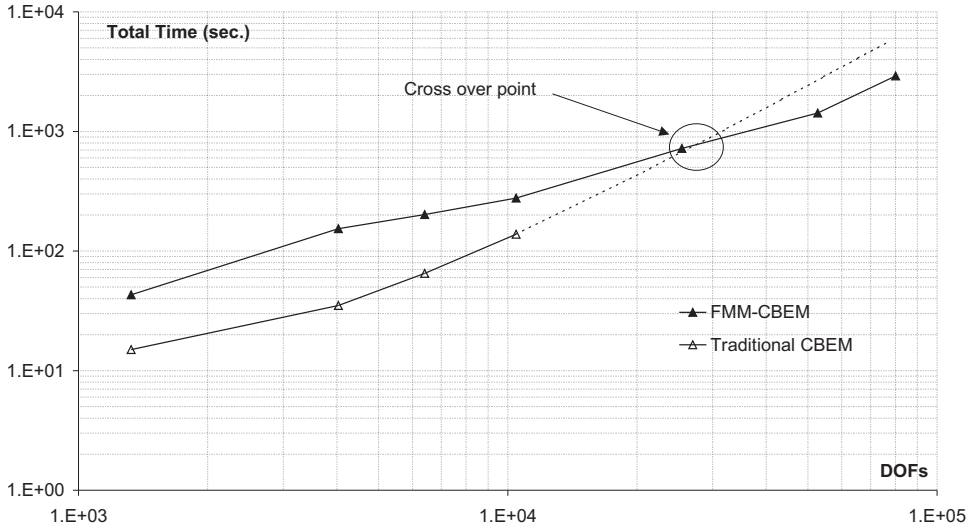

Fig. 7. Inclusion problem: analysis time $T(N)$ against problem size N for the traditional and FMM-based CBEM-FEM coupling methods.

Table 3
Inclusion problem: CPU timing using diagonal preconditioning of BEM equations.

Mesh	1	2	3	4	5	6	7
Iters	48	51	53	55	57	58	60
Total time (s)	63	156	208	315	835	1715	3426

"Cycle" refers to the overall time spent in one BEM matrix-vector products, while "Direct" stands for the computer time spent in reading the sparse direct influence matrix (see end of section 3) and performing the matrix-vector product with the trial solution. The depth of the oct-tree structure was such that a leaf should contain at least $M = 25$ elements. Setting up the preconditioners appears to require about 25% of the overall analysis computing time for all meshes, largely because computing the SPAI of (31) is a relatively time-consuming operation (note in passing that the recorded SPAI computation times are roughly $O(N)$, as expected). The overall analysis time $T(N)$ also appears to increase in a roughly linear fashion with N. In addition, $T(N)$ obtained using either conventional or FMM-based FEM-BEM coupling is plotted against N in Figure 7. The conventional FEM-BEM coupling entails a $O(N_{\text{BEM}}^2)$ storage requirement which prevented to actually perform the analyses for meshes 5, 6 and 7. Hence, this part of the graph, materialized with a dashed line, is the $O(N_{\text{BEM}}^2)$ extrapolation of the values of $T(N)$ recorded for meshes 1 to 4. A cross-over point appears around $N_{\text{c/o}} \approx 3\,10^4$. The value of $N_{\text{c/o}}$ is however expected to be strongly problem-, mesh- and implementation-dependent. In most cases, the total computing time reported in Table 2 exceeds Precond $+$ Cycle $\times$ Iters because of other tasks which are not reported in the table (with the exception of mesh 4, most likely as a consequence of elementary timings being rounded off).

The same problem has been run for the seven meshes using a simpler preconditioning strategy involving a diagonal preconditioning of the BEM equations, i.e. with $[P^{\text{L}}]$ defined by

$$P^{\text{L}} = \begin{bmatrix} \text{Id} & 0 & 0 & 0 \\ 0 & \text{Id} & 0 & 0 \\ 0 & 0 & \text{Id} & 0 \\ 0 & 0 & 0 & \mathcal{C}^{-\text{T}} \end{bmatrix}$$

while the definition (30) for $[P^{\text{R}}]$ is retained. Table 3 displays, for each of the seven meshes, the overall computing time spent and the iteration count. A comparison of tables 2 and 3 reveals that the SPAI-based preconditioning lears to reduced overall analysis times and stable iteration counts, whereas the iteration counts entailed by the simpler preconditioner are higher and slowly increase with the problem size. The sharp increase of "Direct" CPU time between meshes 6 and 7 is likely due to the fact that in the latter case the sparse direct influence matrix could not be held in the core memory.

Finally, to further demonstrate that the FMM-based CBEM-FEM coupling procedure can be applied to large problems, an array of $8 \times 8 \times 8 = 256$ identical spherical inclusions (diameter 1 m, centers placed at the nodes of a regular cubical grid with node intervals of 3 m along the three principal directions) em-

Table 4
Many-inclusion problem: discretization parameters.

Nodes	Elements		Oct-tree		DOFs		
	BEM	FEM	Max. level	Leaves	N_{BEM}	N_{F}	$N = N_{\mathrm{BEM}} + N_{\mathrm{F}}$
93227	122880	326493	5	7176	374784	92289	467073

Table 5
Many-inclusion problem: CPU timing.

Precond. (s)		Time (s)				Iters	Total time
BEM	FEM	Upw.	Downw.	Direct	Cycle	n	(s)
6609	19	47	48	84	180	147	39656

bedded in the infinite elastic medium, subjected to uniform thermal load, has been considered. The other physical parameters are the same as above. The BEM-FEM mesh, whose main characteristics are reported in Table 4, features 480 three-noded triangular boundary elements on each inclusion-matrix interface, about 1200 linear four-noded tetrahedral domain elements in each inclusion, and $N = 467073$ DOFs overall. The many-inclusion aspect of this problem makes it especially well-suited to FMM, which is here used for the evaluation of geometrically well-separated contributions (interaction terms between distinct inclusions). In this example, the truncation parameter $p = 5$ was used in the FMM expansions and the depth of the oct-tree structures was such that a leaf should contain at least 100 elements. Only $m = 25$ non-zero terms for every row have been adopted for the construction of the sparsified preconditioner. Figure 8 displays a vertical section of the array together with the displacement magnitude displacement. The recorded computing times and iteration count are presented in Table 5 in the same format previously used for the single-inclusion problem.

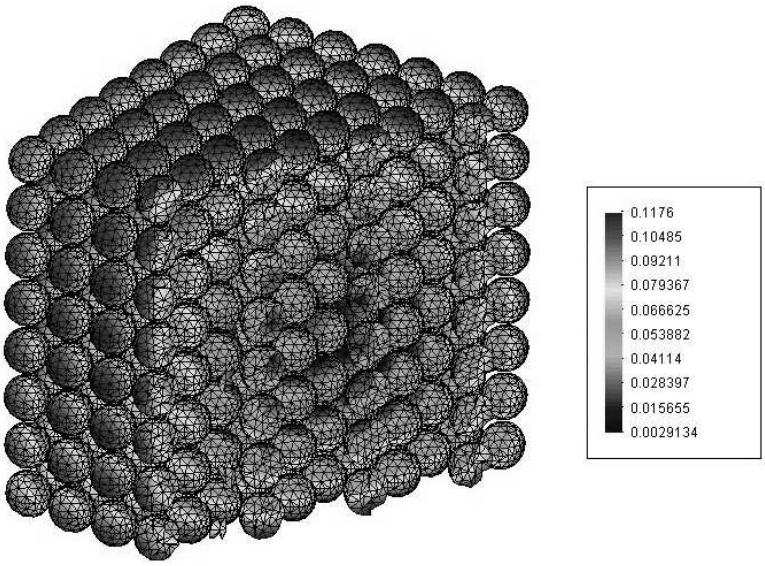

Fig. 8. Many-inclusion problem: displacement magnitude drawn on a vertical section of the array of spherical inclusions

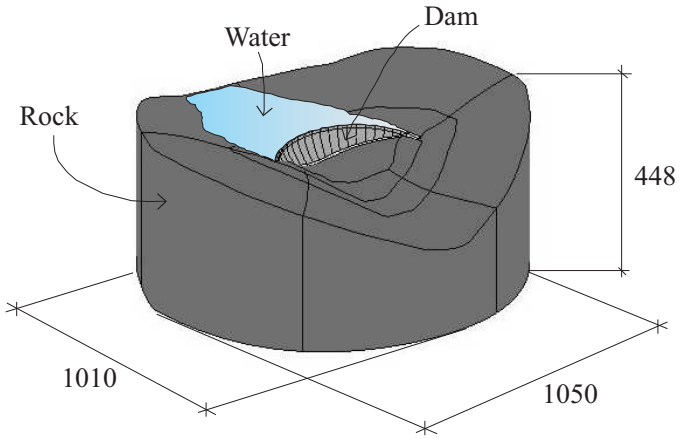

Fig. 9. Solid model of the Pian Telessio dam and rock foundation

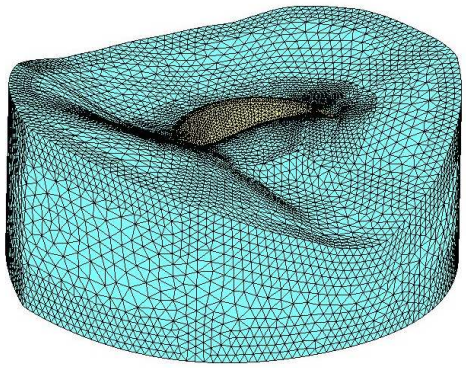

Fig. 10. Pian Telessio dam: mesh 3.

5.2 Pian Telessio dam

This second example involves the Pian Telessio dam, an arc-gravity concrete structure located in the north-west of Italy. The main characteristics of the dam are: maximum height 80 m, crest lenght 515 m, crest maximal altitude 1919 m above sea level, maximum storage level 1917 m above sea level, reservoir volume $23.5\,10^6$ m^3. The structure and a large portion of the surrounding rock foundation around the dam have been subdivided into three zones, as sketched in Figure 9: the dam, the concrete layer, and the rock foundation. The boundary conditions are

Table 6
Pian Telessio dam: discretization parameters for the three CBEM-FEM meshes (1,2,3) and the ABAQUS FEM comparison model (A).

Mesh	Nodes	Elements		Oct-tree		DOFs		
		BEM	FEM	Max. level	Leaves	N_{BEM}	N_{F}	Total
1	25443	21684 (T3)	73569 (T4)	9	8953	38118	43797	81915
2	23433	7726 (T6)	10307 (T10)	8	3548	50490	46773	97263
3	51978	15296 (T6)	14462 (T10)	8	6786	96636	64152	160788
A	406035	—	279742	—	—	—	—	1218105

Table 7
Pian Telessio dam: CPU timing for the three CBEM-FEM meshes (1,2,3) and the ABAQUS FEM comparison model (A).

Mesh	Precond. (s)		Time (s)				Iters	Total time
	BEs	FEs	Upw.	Down.	Direct	Cycle	n	(s)
1	186	26	23	27	24	76	83	7916
2	328	114	11	21	23	57	82	5818
3	1215	223	23	36	102	165	85	17775
A	—	—	—	—	—	—	—	3749

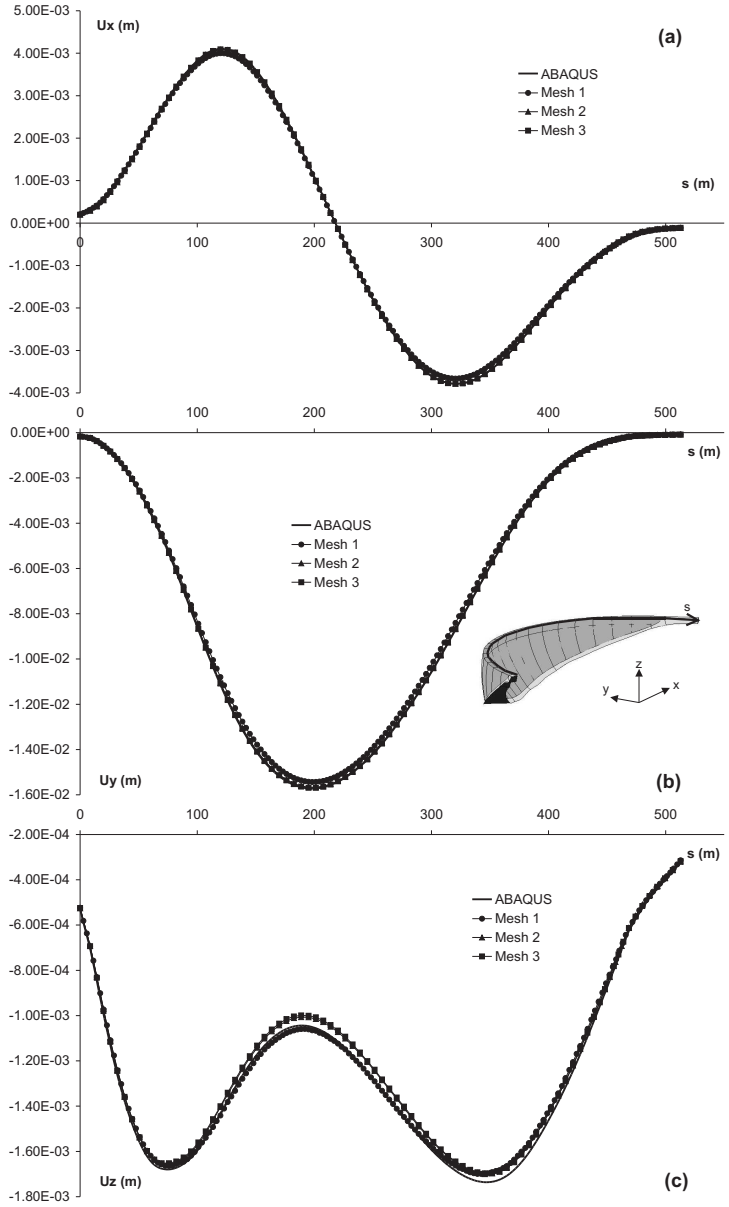

Fig. 11. Pian Telessio dam: crest displacement Ux (a), Uy (b) and Uz (c)

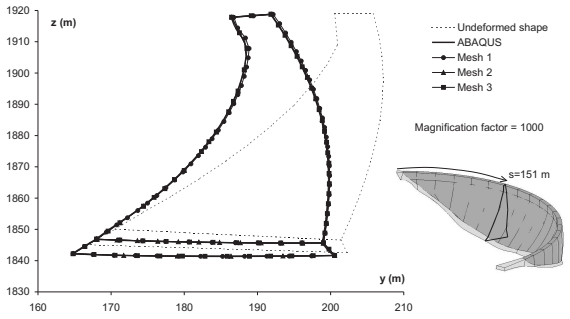

Fig. 12. Pian Telessio dam: deformed shapes of a transversal section

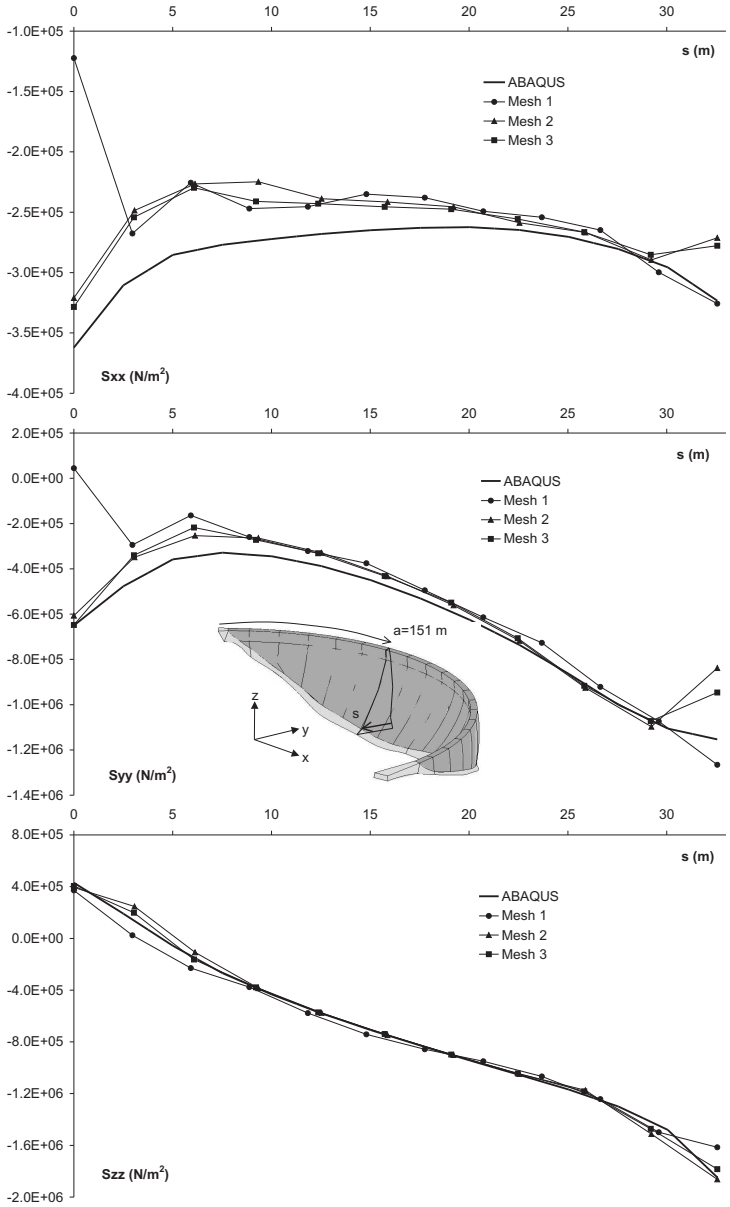

Fig. 13. Pian Telessio dam: normal stress components

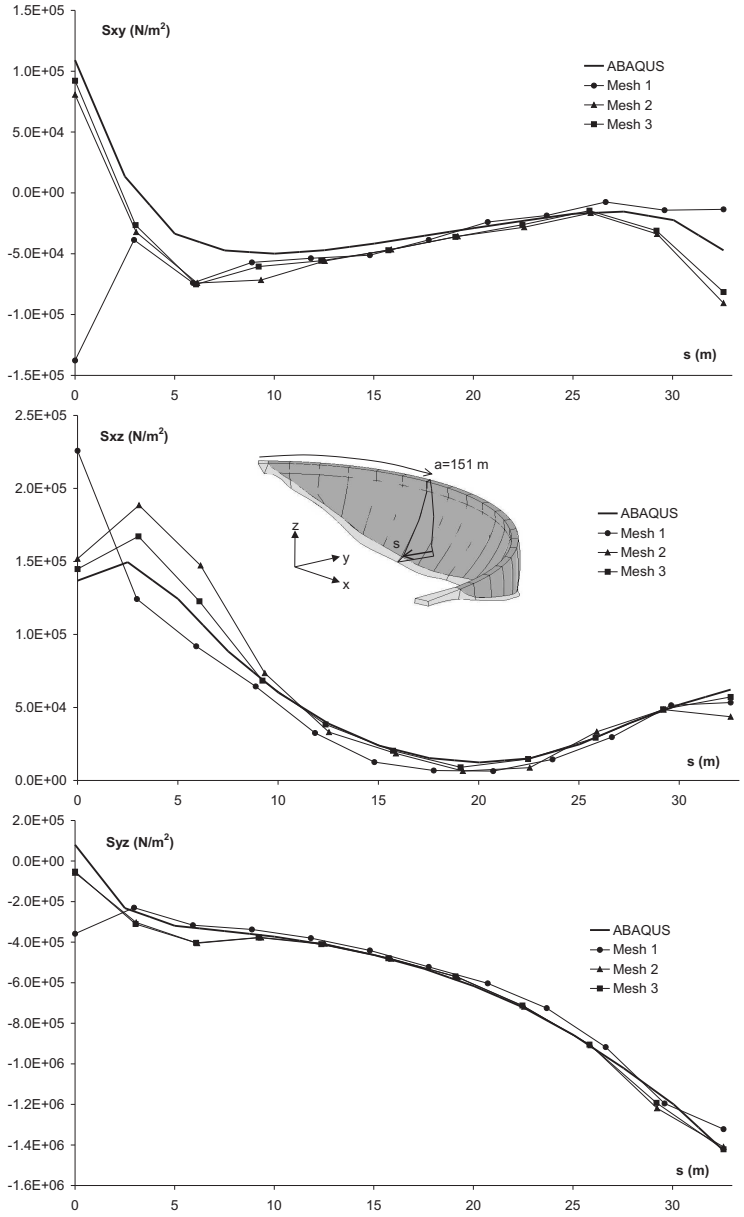

Fig. 14. Pian Telessio dam: shear stress components

as follows: zero displacements are imposed on the bottom and lateral surfaces of the rock-foundation subdomain, a hydrostatic pressure is applied on the rear face of the dam to take into account the stored water, and the other external surfaces are traction-free. Finite elements were used for the first two zones, and boundary elements for the third zone. All materials are treated as linear elastic, with the following characteristics: $E = 33000$ MPa, $\nu = 0.12$ and $\gamma = 2500$ Kg/m^3 (dam), $E = 30000$ MPa, $\nu = 0.12$ and $\gamma = 2500$ Kg/m^3 (concrete layer) and $E = 20000$ MPa, $\nu = 0.2$ (rock). Three CBEM-FEM meshes of increasing sizes have been set up, together with a FEM model run with ABAQUS for comparison purposes. Mesh 1 is based on linear three-noded boundary triangular elements and four-noded tetrahedral domain elements, while meshes 2 and 3 are based on

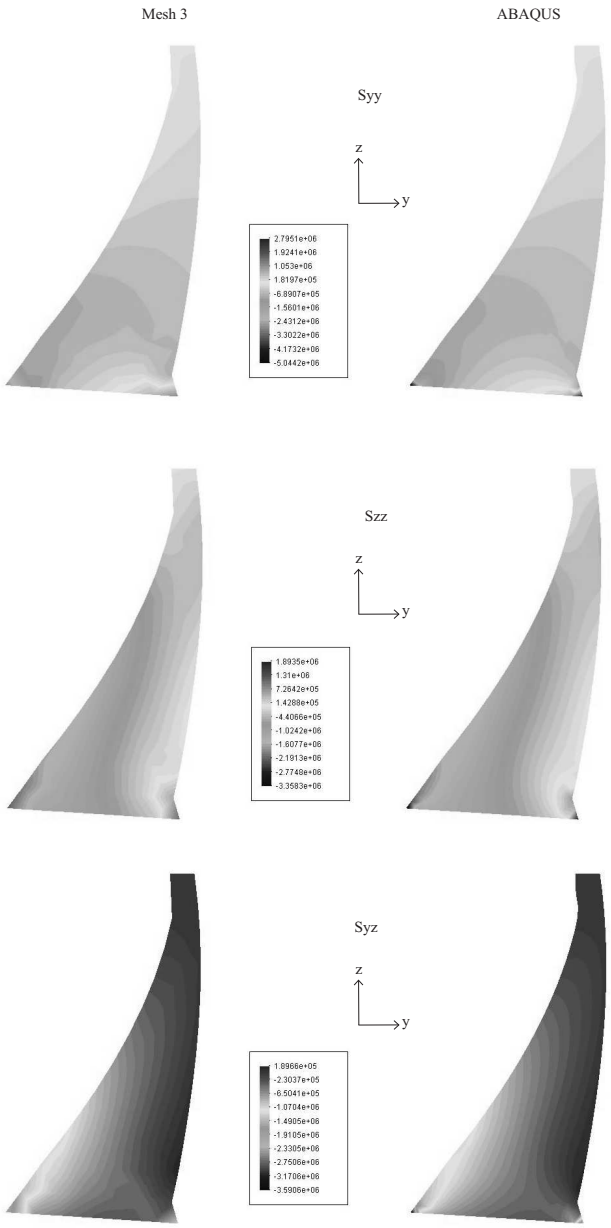

Fig. 15. Pian Telessio dam: stress components on the vertical section (crest abscissa equal to $151m$) obtained with the third mesh (left) and ABAQUS (right).

quadratic six-noded triangular boundary elements and ten-noded tetrahedral domain elements. Figure 10 depicts mesh 3, which involves about $N_{\text{BEM}} = 10^5$ and $N_{\text{F}} = 6\,10^4$ DOFs, respectively.

The truncation parameter in multipole expansions and the maximum number of elements in a leaf have been set to $p = 10$ and $M = 30$, respectively. GMRES was run without restarts. The SPAI preconditioning technique presented in section 4 has been adopted, the sparsified matrix being build with $m = 50$ non-zero coefficients on each row. The complete CC^t factorization has been chosen as left preconditioner for the finite element coefficient matrix, as explained in the previous section.

The iteration counts and CPU times required by the various phases of the FMM and by the overall analyses, recorded for the three CBEM-FEM meshes of Table 6, are gathered in Table 7 using the same arrangement as in Table 2. Again, a sharp increase of "Direct" CPU time between meshes 2 and 3 indicates that sparse direct influence matrix for mesh 3 could not be held in the core memory.

The crest displacements computed using the present CBEM-FEM technique, displayed in Figure 11, are seen to agree well with values obtained using the FEM code ABAQUS. Discrepancies between the two computations have been registered for the u_z component, with a maximum difference of approximately $4{\cdot}10^{-5}$ m when using mesh 3. Obviously, if a better accuracy is required the solution procedure should be stopped for a smaller relative error.

The deformed shapes of a transversal section of the dam (at a crest abscissa of 151 m), depicted in Figure 12, are also in good agreement with the ABAQUS solution.

In Figures 13 and 14, the stress components evaluated on the interface between the concrete layer and the dam, along the s path, are reported. Meshes 2 and 3 (based on quadratic elements) yield better results than mesh 1, based on linear tetrahedral elements which are not well suited for highly accurate stress analyses.

6 Summary

In this paper a coupling technique between the finite element method (FEM) and the collocation boundary element method (CBEM) has been presented. Its main feature is the recourse to the Fast Multipole Method in order to both accelerate the BEM contribution to the overall computation and allow BEM meshes involving large numbers of degrees of freedom. In addition, a preconditioning strategy based on the sparse approximate inverse concept has been implemented. All these issues have been demonstrated and validated through numerical experiments, run on a Linux PC computer, involving up to about $4\,10^5$ BEM unknowns. The accuracy of the numerical solutions is not influenced by the FMM if certain parameters (multipole truncation parameter p, minimum number M of elements in a leaf cell) are set appropriately.

Several implementation areas are believed to have potential for improvement of the overall computational performance. The heuristic for selecting the sparsity pattern in the SPAI algorithm was a rather simple one, and can certainly be improved upon. The preconditioning of the FEM part (a full Choleski decomposition of the FEM stiffness matrix) certainly can be improved upon, with potential savings in core memory space. The computation and handling of the sparse direct influence matrix also has scope for optimization.

The CBEM-FEM coupling strategy was presented here for overall linear problems. However, the extension of the present iterative algorithm to problems involving constitutive nonlinearity (e.g. plasticity) in the FEM domain should not give rise to major difficulties.

References

[1] ALIABADI, M. H. Boundary element formulations in fracture mechanics. *Appl. Mech. Rev.*, **50**, 83–96 (1997).

[2] ALLEON, G., BENZI, M., GIRAUD, L. Sparse approximate inverse preconditioning for dense linear system arising in computational electromagnetics. *Numerical Algorithms*, **16**, 1–15 (1997).

[3] BEER, G. *Programming the boundary element method*. John Wiley and Sons (2001).

[4] BENZI, M., TUMA, M. A sparse approximate inverse preconditioner for nonsymmetric linear system. *SIAM J. Sci. Comput.*, **19**, 968–994 (1998).

[5] BONNET, M. *Boundary integral equation method for solids and fluids*. Wiley (1999).

[6] FRANGI, A., NOVATI, G., SPRINGHETTI, R., ROVIZZI, M. 3D fracture analysis by symmetric Galerkin BEM. *Computational Mechanics*, **28**, 220–232 (2002).

[7] GANGULY, S., LAYTON, J.B., BALAKRISHNA, C. Symmetric coupling of multi-zone curved Galerkin boundary elements with finite elements in elasticity. *Int. J. Numer. Meth. Engng.*, **48**, 633–654 (2000).

[8] GREENBAUM, A. *Iterative methods for solving linear systems*. SIAM, Philadelphia, USA (1997).

[9] GREENGARD, L., ROKHLIN, V. A fast algorithm for particle simulations. *J. Comp. Phys.*, **73**, 325–348 (1987).

[10] GREENGARD, L., ROKHLIN, V. A new version of the fast multipole method for the Laplace equation in three dimensions. *Acta Numerica*, pp. 229–269 (1997).

[11] GROTE, M. J., HUCKLE, T. Parallel preconditioning with sparse approximate inverses. *SIAM J. Sci. Comput.*, **18**, 838–853 (1997).

[12] GUIGGIANI, M., GIGANTE, A. A general algorithm for multidimensional cauchy principal value integrals in the boundary element method. *Journal of Applied Mechanics*, **112**, 906–915 (1990).

[13] HAAS, M., KUHN, G. Mixed-dimensional, symmetric coupling of FEM and BEM. *Engng. Anal. Boundary Elements*, **27**, 575–582 (2003).

[14] HSIAO, G.C., SCHNACK, E., WENDLAND, W.L. A hybrid coupled finite-boundary element method in elasticity. *Comput. Methods Appl. Mech. Engrg.*, **173**, 287–316 (1999).

[15] LI, H.B., HAN, G.M., MANG, H.A., TORZICKY, P. A new method for the coupling of finite element and boundary element discretized subdomains of elastic body. *Comp. Meth. Appl. Mech. Engng.*, **54**, 161–185 (1986).

[16] MANTIC, V. A new formula for C-matrix in the Somigliana identity. *Journal of Elasticity*, **33**, 191–201 (1993).

[17] MURA, T. *Micromechanics of defects in solids*. Martinus Nijhoff Publishers (1987).

[18] NISHIMURA, N. Fast multipole accelerated boundary integral equation methods. *Applied Mechanics Review*, **55, n. 4**, 299–324 (2002).

[19] POLIZZOTTO, C., ZITO, M. Variational formulations for coupled BE/FE methods in elastostatics. *ZAMM Z angew. Math. Mech.*, **11**, 533–543 (1994).

[20] ROKHLIN, V. Rapid solution of integral equations of classical potential theory. *J. Comp. Phys.*, **60**, 187–207 (1985).

[21] SAAD, Y. *Iterative methods for sparse linear systems*. SIAM, Philadelphia, USA (2003).

[22] SAAD, Y., SCHULTZ, M. H. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, **7**, 856–869 (1986).

[23] YOSHIDA, K. *Application of fast multipole method to boundary integral equation method*. Ph.D. thesis, Dept. Global Environment Eng., Kyoto Univ., Japan (2001).

[24] YOSHIDA, S., NISHIMURA, N., KOBAYASHI, S. Application of fast multipole Galerkin boundary integral equation method to elastostatic crack problems in 3D. *Int. J. Num. Meth. in Eng.*, **50**, 525–547 (2001).

[25] ZENG, X., KALLIVOKAS, L.F., BIELAK, J. A symmetric variational finite element-boundary integral equation coupling method. *Computers and Structures*, **46**, 995–1000 (1993).

A Fast Multipole Method: useful formulae

This appendix provides the formulae for practical computation of the solid harmonics $R_{n,m}$ and $S_{n,m}$, and of the derivatives of the $R_{n,m}$, after references [23, 24] to which the reader is referred for more details. First, the evaluation of $R_{n,m}(\boldsymbol{z})$ and $S_{n,m}(\boldsymbol{z})$, where $\boldsymbol{z}$ denotes the generic argument of these functions, is performed as follows:

- The $R_{n,m}(\boldsymbol{z})$ are computed recursively by setting $R_{0,0}(\boldsymbol{z}) = 1$ and using

$$R_{n+1,n+1}(\boldsymbol{z}) = \frac{z_1 + \mathrm{i}z_2}{2(n+1)} R_{n,n}(\boldsymbol{z}) \tag{A.1}$$

$$((n+1)^2 - m^2) R_{n+1,m}(\boldsymbol{z}) - (2n+1) z_3 R_{n,m}(\boldsymbol{z}) + |\boldsymbol{z}|^2 R_{n-1,m}(\boldsymbol{z}) = 0 \tag{A.2}$$

- The $S_{n,m}(\boldsymbol{z})$ are computed recursively by setting $S_{0,0}(\boldsymbol{z}) = 1/|\boldsymbol{z}|$ and using

$$S_{n+1,n+1}(\boldsymbol{z}) = \frac{(2n+1)(z_1 + \mathrm{i}z_2)}{|\boldsymbol{z}|^2} R_{n,n}(\boldsymbol{z}) \tag{A.3}$$

$$|\boldsymbol{z}|^2 S_{n+1,m}(\boldsymbol{z}) - (2n+1) z_3 S_{n,m}(\boldsymbol{z}) + (n^2 - m^2) S_{n-1,m}(\boldsymbol{z}) = 0 \tag{A.4}$$

- Finally, $R_{n,m}(\boldsymbol{z})$ and $S_{n,m}(\boldsymbol{z})$ for negative values of m are computed via the identities

$$R_{n,-m}(\boldsymbol{z}) = (-1)^m \overline{R_{n,m}(\boldsymbol{z})} \qquad S_{n,-m}(\boldsymbol{z}) = (-1)^m \overline{S_{n,m}(\boldsymbol{z})} \tag{A.5}$$

Identities (A.2) and (A.4) are used with $n \geq m$. In addition, the derivatives of the $R_{n,m}$ are computed by means of the following formulae:

$$\begin{aligned}
\frac{\partial}{\partial z_1} R_{n,m}(\boldsymbol{z}) &= \frac{1}{2}(R_{n-1,m-1} - R_{n-1,m+1})(\boldsymbol{z}) \\
\frac{\partial}{\partial z_2} R_{n,m}(\boldsymbol{z}) &= \frac{i}{2}(R_{n-1,m-1} + R_{n-1,m+1})(\boldsymbol{z}) \\
\frac{\partial}{\partial z_3} R_{n,m}(\boldsymbol{z}) &= R_{n-1,m}
\end{aligned} \tag{A.6}$$